

AF/2124  
PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE



Appl. No. : 10/037,390  
Applicant : Wilkinson et al.  
Filed : 10/23/2001  
TC/A.U. : 2124  
Examiner : Chavis, John Q.

Confirmation No. 7729

#17  
J. Cant  
6/9/04

Docket No. : 40.0010 C1  
Customer No. : 26751

Certificate of Mailing  
I hereby certify under 37 CFR 1.8 that this correspondence  
is being deposited with the United States Postal Service with  
sufficient postage as first class mail in an envelope addressed to:  
Mail Stop Appeal Brief-Patent, Commissioner for Patents, P.O.  
Box 1450, Alexandria, VA 22313-1450 on May 24, 2004.

Typed or printed name of person signing this certificate:

Cindy Thane  
Signature: Cindy Thane

Mail Stop Appeal Brief-Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

RECEIVED  
MAY 28 2004  
Technology Center 2100

REPLY BRIEF

Noted  
8-10-04  
CCE  
Appellants have invented how to develop applications written in a high-level  
language and to have those applications execute on a resource-constrained device such as  
a micro-controller, and consequently, micro-controllers that have been thus programmed.  
This invention is exemplified by Claim 106 which can be summarized as follows:

“A microcontroller, comprising:

a memory storing an application derived ... by ... compiling ... converting the  
compiled form into a converted form; and

an interpreter configured to interpret derivative applications in the converted form

...”

The prior art (Peyret and Renner) cited by the Examiner does not teach or suggest such a micro-controller.

In the Examiner's Answer, the Examiner appears to question, with respect to the claim elements that deal with conversion of compiled form, the scope to which the Appellants are entitled by stating that "the claims merely mention 'an interpreter configured to interpret applications in the converted form'" (Examiner's Answer, Page 3, lines 20-21) as opposed to being directed to a specialized converter. That the applicants have invented how to put a high-level language on a smart card is beyond dispute (the parent application has already issued into a patent, U.S. Patent No. 6,308,317). Thus, one issue for this case is whether the prior art is such as to limit the Appellants to the claims of that patent or is entitled to broader claims, for example, an interpreter for particular conversions or a more general one as claimed.

The Examiner states that "Standard Java is considered to provide for this feature [i.e., configured to interpret applications in the converted form] when code is compiled (sic) on one system and then transferred to another" (Examiner's Answer, Page 3, line 21 – Page 4, Line 2). That is not true. In fact, Java was designed with the opposite in mind. Java was designed to enable "Write Once, Run Anywhere" so that a Java program that is compiled may run on a Java Virtual Machine on any computer. See for example, <http://java.sun.com/features/1998/01/wora.html>. Thus, "Standard Java" teaches away from the proposition that the Examiner asserts that "Standard Java is considered to provide."

Therefore, the Examiner's statement that "the converting occurs because each system utilizes its own virtual machine" is nonsensical. It is *because* that each system utilizes its own virtual machine that conversions are not necessary for Java programs.

The Examiner goes on to say that Peyret's system utilizes a reduced interpreter (which further suggests that some type of converting occurs between the original system (CPU, with the standard interpreter) and the smartcard (with the reduced interpreter or specialized interpreter)" (Examiner's Answer, Page 4, Lines 5-8). This conclusion is based on several incorrect premises. First, there is no disclosure of a standard interpreter on the first system (server 82, Peyret, Fig. 4). Peyret is concerned with loading

applications stored on a server onto a smart card. Since usually smart card applications do not run on conventional computers, no one would assume that they would run on the server. Therefore, there would not be a need for two versions on the applications.

Second, Peyret does not ever describe how his applications are written. Therefore, one would assume that they are written for the target system (the smart card with its reduced interpreter) because that is how computer programs normally are written. For these two reasons, one would assume that Peyret did not need or contemplate a converter. More importantly, to a person of ordinary skill in the art, Peyret would not suggest the need or benefit of a converter and it is not inherent in his disclosure.

In the Examiner's Answer, the Examiner cites various passages from the Specification to explain what "the converted code is in applicant's invention" and he cites compressing files, and converting from specific bytecodes to generic bytecodes. The Examiner then makes the assertion that "[t]his is considered the essence of what Peyret does in transferring code from a CPU system (one system) onto a smartcard (the second system), see the abstract and fig. 4." (Examiner's Answer, Page 4, second paragraph – Page 5, line 2). That assertion is a manifestly incorrect statement.

Fig. 4 of Peyret (attached hereto as Appendix A) is indeed a suitable illustration for demonstrating some of the differences between Appellants' invention and the disclosure of Peyret. Peyret's Fig. 4 shows a smart card 20 having loaded thereon two applications 94 and 96. The smart card is connected to a Server 82 via a terminal 80. The server 82 has stored thereon a couple of applications 98 and 100. The applications stored on the server 82 are loaded onto the smart card 20 ("Therefore, a new copy of the applet 98 with refreshed use rights, located on the server 82, may be loaded into the NVM of the smart card" (Peyret, Col. 7, lines 63-65) and "a new 100 applet having use rights may be loaded into the smart card 20 from the server 82" (Peyret, Col. 8, lines 1-3). Applicants do not dispute that Peyret teaches loading applications from a server onto a smart card. However, Peyret does not describe the structure or source of the applications 98 and 100 stored on the server. Therefore, it is not surprising that Peyret fails to teach or suggest that the applications have been "derived from an application having a class file format wherein the application is derived from an application having a

class file format by first compiling the application having a class file format into a compiled form and then converting the compiled form into a converted form” (Claim 106).

Contrast this with Figure 2 of Applicants’ patent application (attached hereto as Appendix B). In Figure 2, Appellants illustrate that programs in the form of a card class file 27 are loaded onto an Integrated Circuit Card 10. But that is not all. Figure 2 also illustrates that the card class file 27 have been derived from the two step process of processing a Java Application 20 through a Java Application Development Environment 22 (in the parlance of Claim 106, “first compiling the application having a class file format into a compiled form”) and then converting the resulting application class files 24 using the card class file converter 26 into the card class file 27 (In the parlance of Claim 106, “then converting the compiled form into a converted form”).

Thus while there is some similarity between the loading of application programs 98 and 100 into the smart card 20 (Peyret) and the loading of card class file 27 onto the integrated circuit card 10 (Fig. 2, Applicant’s), the elements of the claims that correspond to the elements 20, 22, 24, and 26 of Applicant’s figure 2 is missing from Peyret. There is nothing in Peyret that comes close to teaching or suggesting those claimed elements.

The Examiner relies on Renner for that missing teaching. However, there is no suggestion to combine Renner and Peyret and such a combination would not result in Appellants’ claimed invention.

Appellants take exception to the Examiner’s combination of Renner with Peyret. As discussed in the Appeal Brief, Renner (5,679,945) teaches a system in which smart cards can be used to replace magnetic stripe readers, bar code readers, and Wiegand effect readers. The Examiner makes the erroneous statement that “applicant’s page 4 lines 11-14 admitted prior art, further justifies the combining of Peyret/Renner to conserve memory” (Examiner’s Answer, Page 5, line 4-5). Appellants’ point out in the quoted passage that “because of the constrained environment, applications for smart cards are typically written in a low level programming language (e.g., assembly language) to conserve memory. How that would justify combining a reference for replacing smart

cards for mag stripe readers, etc., (Renner) with a reference for loading applications onto smart cards (Peyret) is something that goes beyond the wildest imagination. More importantly, Renner teaches the conversion between data formats used by smart cards into formats used by Renner's various target devices (e.g., mag stripe readers and bar code readers). It must be evident to anyone that such a conversion is vastly different from the conversion element of Appellant's claim. Taking Renner's conversion step and adding it to Peyret while not suggested by either reference, would not come close to constituting "converting the compiled form into a converted form" (Claim 106).

The Examiner has made the statement "However, the applicant should note that since microcontrollers (like microcomputers) understands on (sic) ones and zeroes when executing, compiling is considered an inherent feature of Renner's system to enable execution" (Examiner's Answer, Page 5, lines 17-20). "Compiling" is a term of art in the art of Computer Science meaning "To transform a program written in a high-level programming language from *source code* into *object code*" (Webopedia <http://www.webopedia.com/TERM/c/compile.html>). Just because computers operate on ones and zeros does that mean that all data in the computer has gone through a compilation. Appellant's wish to use this very document as an example. Appellant's attorney are writing these words using the Microsoft Word word-processing system and before the Examiner and the board reads these words, Appellant's attorney will have saved this document to disk. On the disk it will be stored as ones and zeros. Yet, this document will not be compiled. Thus, it is wrong to draw the inference that because ones and zeros are used in computers that a computerized data file has been compiled. In the case of Renner, there is no reason to believe that the Renner smart card codes are compiled because the target devices are mag stripe readers, bar code readers and Wiegand effect readers. These devices do not expect computer programs.

Thus, not only is there no motivation to combine Peyret and Renner, the proposed combination would not be equivalent to Appellants' invention because both Peyret and Renner fail to teach or suggest "the application is derived from an application having a class file format by first compiling the application having a class file format into a

compiled form and then converting the compiled form into a converted form" (Claim 106).

Conclusion

For the reasons provided in the Appeal Brief and further explained herein, it is submitted that the claims now before the Board of Appeals are allowable. Accordingly, Appellants respectfully request reversal of the rejections set forth by the Examiner and early allowance of the Application.

Respectfully submitted,



Pehr B. Jansson  
Registration No. 35,759

Date: May 24, 2004

Enclosures:

1. Return Receipt Postcard
2. This Appeal Brief (6 pages)
3. Appendix A: Figure 4 from Peret (5,923,884)
4. Appendix B: Figure 2 from Application 10/037,390

Customer No. 41754  
Pehr Jansson  
7628 Parkview Circle  
Austin, TX 78731  
Tel: 512-241-0837  
Fax: 678 868 0101

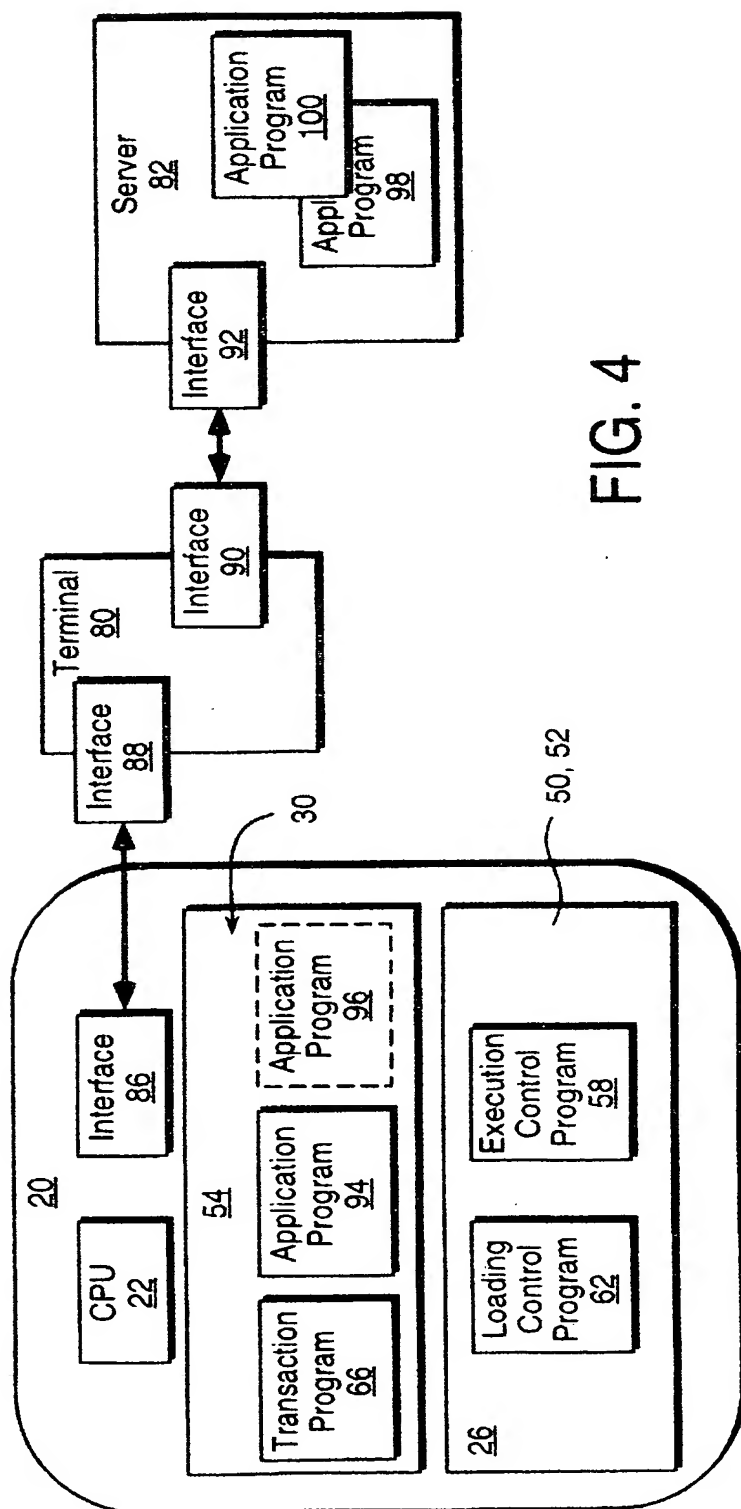


FIG. 4

2/23

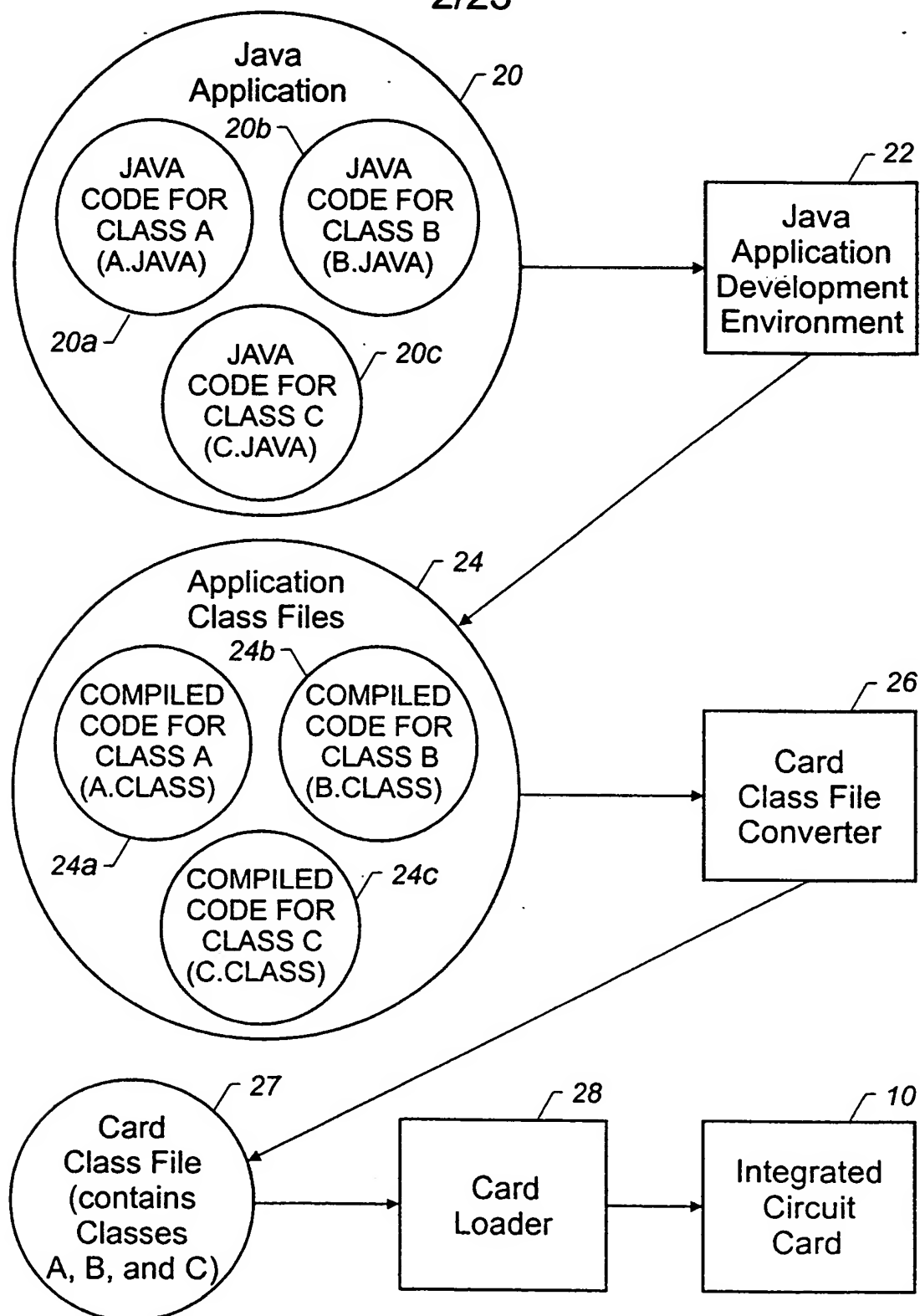


FIGURE 2



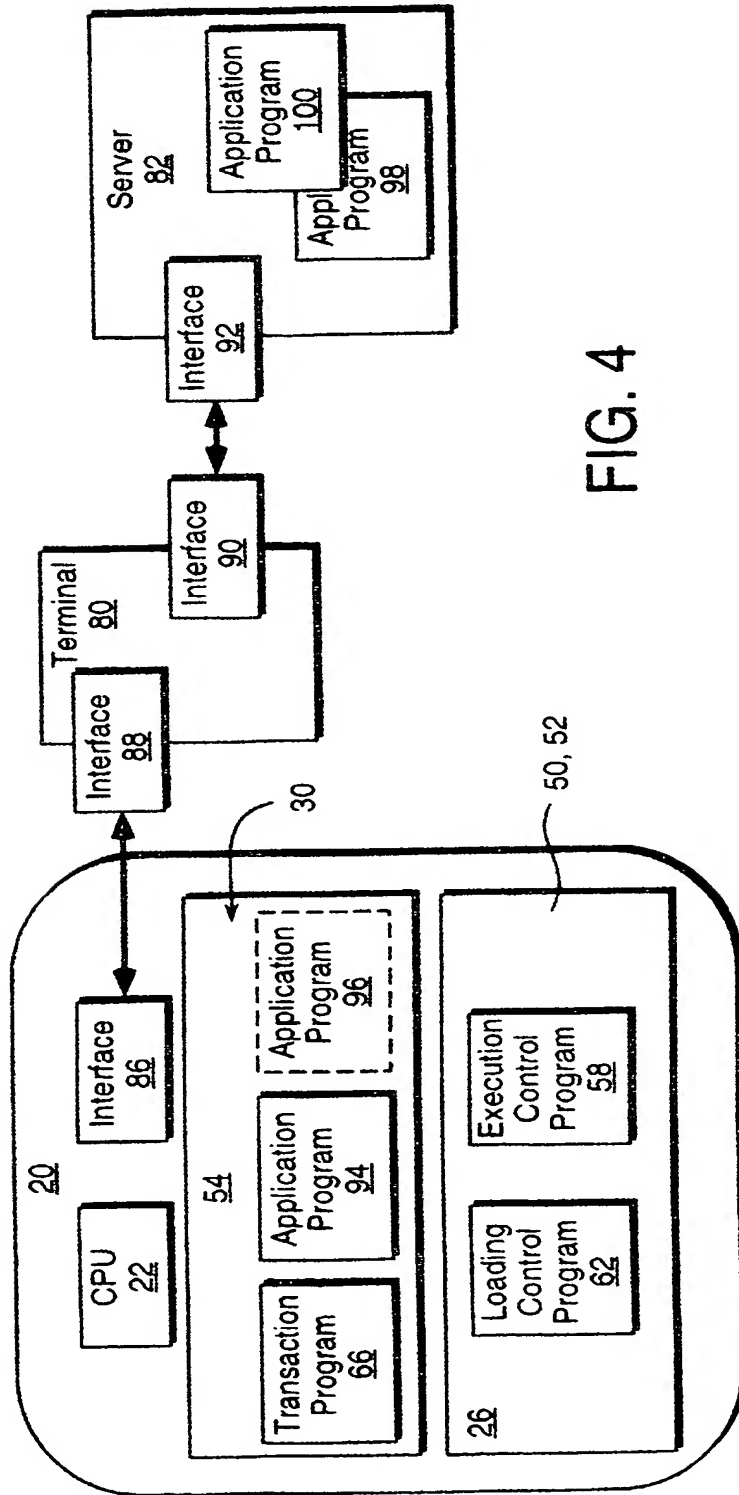


FIG. 4

APPENDIX B:  
FIGURE 2 OF APPLICANTS' PATENT APPLICATION (10/037,390)

2/23

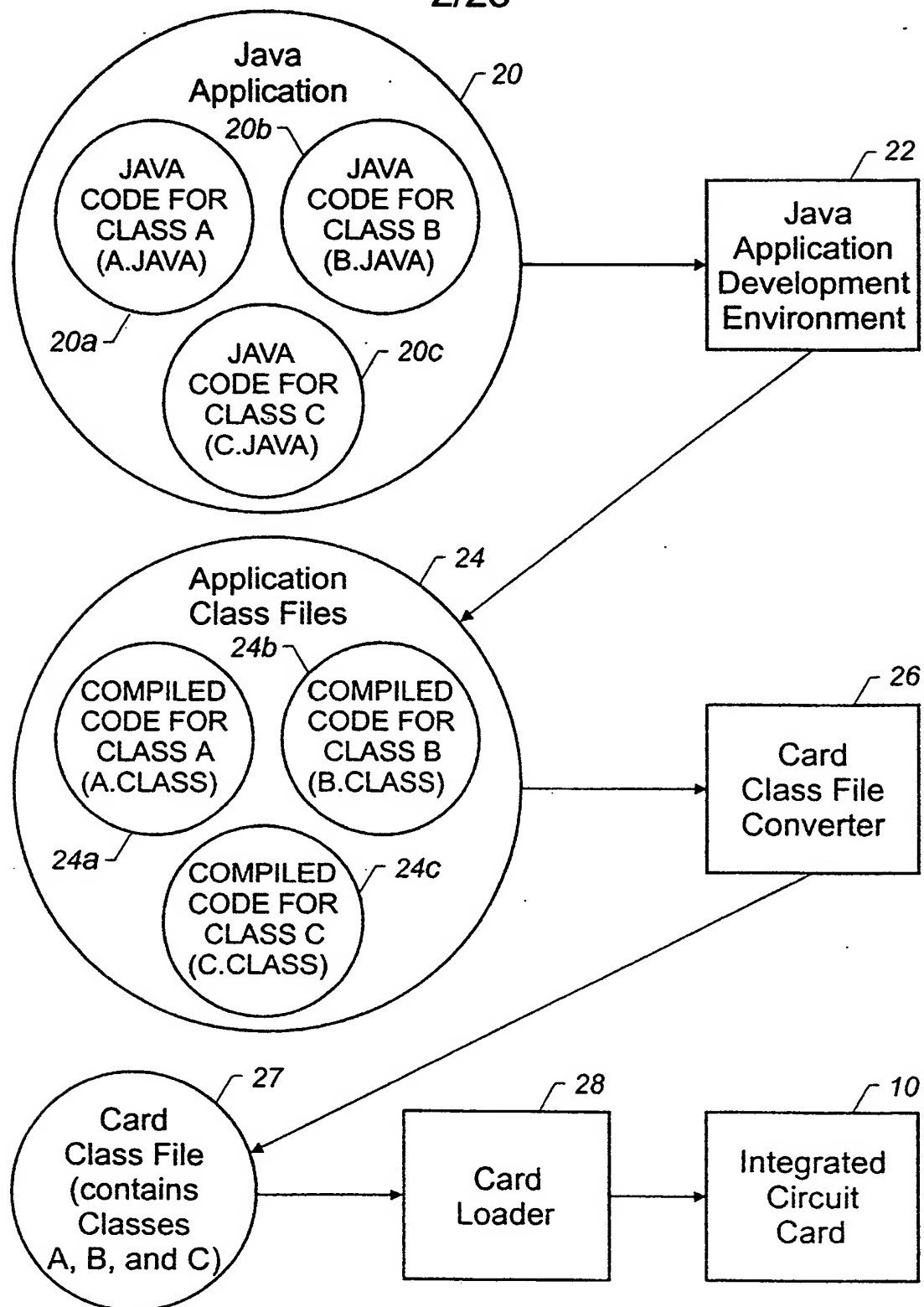


FIGURE 2

**Patent Attorney Docket No. 40.0010 C1**

The Patent Office date stamp sets forth the receipt date for the following correspondence:

**Applicant: Wilkinson, et al**

**Mailed on May 24, 2004 by Regular Mail to Mail Stop Appeal Brief-Patents,  
Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

1. Postcard for PTO Date Stamp and Return
2. Certificate of Mailing (printed on 1<sup>st</sup> page of brief)
3. Reply Brief & Appendices  
(6 pages & 2 pages)

U.S. Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450



Pehr Jansson  
7628 Parkview Circle  
Austin, TX 78731-1127